

The Effects of Domain and Collaboration Feedback on Learning in a Collaborative Intelligent Tutoring System

Jay Holland¹, Nilufar Baghaei², Moffat Mathews¹, Antonija Mitrovic¹

¹Intelligent Computer Tutoring Group, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand.

² Dept of Computing, Unitec Institute of Technology, Auckland, New Zealand.
tanja.mitrovic@canterbury.ac.nz

Abstract. We present initial results from a study comparing the effects of domain and collaboration feedback on learning within COLLECT-*UML*, a collaborative problem-solving ITS. Using COLLECT-*UML*, two students in separate physical locations (a collaborative pair) construct UML class diagrams to solve problems together. In the default version, COLLECT-*UML* provides both domain and collaboration feedback. In this study however, collaborative pairs were randomly assigned to one of four modes (treatment conditions) which varied the feedback presented by the system: no feedback (NF), domain feedback only (DF), collaborative feedback only (CF), and both domain and collaborative feedback (DCF). All conditions improved significantly between pre- and post-test, showing that practicing within COLLECT-*UML* helps learning. At a surface level, collaborative pairs in all modes had similar amounts of collaboration. The DCF mode had significantly higher learning gains than the other modes, indicating the value of receiving both domain and collaborative feedback. Surprisingly, the CF mode had the lowest learning gains (lower than NF), suggesting that, in this case, good collaboration without domain feedback could have simply reinforced erroneous domain knowledge.

Keywords: Collaboration, domain feedback, collaborative feedback

1 Introduction

Researchers in Computer Supported Collaborative Learning have shown the benefits of adaptive collaboration support in Intelligent Tutoring Systems (ITSs) [1-2]. We previously extended COLLECT-*UML* with a collaboration model which provided students with automatic feedback on their collaboration in addition to on-demand domain feedback [3-4]. Here, we present the initial results of a study in which we attempt to separate the effects of domain and collaborative feedback and find their effect on learning.

COLLECT-*UML* is a constraint-based collaborative ITS which provides students with opportunities to practice their Unified Modeling Language (UML) skills by collaborating with a partner [3]. The system automatically creates collaborative pairs

by connecting two students who have logged in and are still unpaired. The web interface provides each student with two solution spaces (individual and group). The intention is that each student first thinks about the problem individually (while creating their individual diagram) before contributing to the shared group diagram. Each student is encouraged to communicate (e.g. discuss their knowledge, provide explanations, seek justifications) with their partner via a chat interface. COLLECT-*UML* stores both domain and collaboration student models and, in the default version, provides students with two types of feedback: domain and collaboration feedback.

2 Evaluation

COLLECT-*UML* has been used in a second-year Software Engineering course (COSC224) at the University of Canterbury for the last few years. We used the lab sessions during the week of 20 September 2010 (week six of the course) to conduct the evaluation study. The intention was to have a setting that was as close to the normal learning environment experienced by students. Seventy-two COSC224 students participated in this study for no reward. None of these students had prior experience with COLLECT-*UML*. Written pre- and post-tests were administered during which students were given ten minutes to answer questions relating to UML diagrams. Both tests were comparable in difficulty. Following the pre-test, students were asked to read a one-page document which contained basic instructions for the study and guidelines for good collaboration [4].

Each collaborative pair was randomly placed into one of four treatment conditions (modes). Each mode altered the type of feedback students received: 1) no feedback (NF), 2) domain feedback only (DF), 3) collaboration feedback only (CF), and 4) domain and collaboration feedback (DCF). Students who received domain feedback (DF and DCF) could submit their solutions at any time to get feedback. Students who did not receive any domain feedback (NF and CF) were instructed to work on their problems till the pair jointly agreed that the solution was correct before moving on to another problem. All modes could request to view the full solution. However, as the full solution is a form of domain feedback, all students were advised that viewing the full solution would lock their problem (i.e. they would not be able to continue working on the problem after viewing the full solution). The system logged all actions performed, including their chats. The system regularly updated all student collaboration models; however, only modes CF and DCF received feedback on their collaboration. All other aspects of the system were identical between modes.

Sixty-one students completed both tests (Table 1). There were no significant differences on the pre-test. However, all modes improved significantly between pre- and post-test (all with $p < 0.01$). DCF had significantly higher gain than the other modes ($F = 4.46$, $p < 0.01$), even when the normalized gain is used ($F = 3.48$, $p = 0.02$); conversely, CF had the lowest gain.

The number of times a student held the pen (to modify the group solution), the chat file size, and the number of changes made to the solution are shown in Table 1. These give us an idea of the amount of collaboration at a surface level. There were no

significant differences between the modes indicating that the amount of collaboration was relatively similar between groups. However, further analyses have to be conducted to examine the quality of these collaborative actions.

Table 1: Statistics for all treatment groups

Mode (# students)	NF (20)	DF (18)	CF (18)	DCF (16)
Test completed	19	12	15	15
Pretest	2.4 (0.9)	2.3 (0.8)	2.4 (0.8)	1.7 (1.1)
Posttest	3.9 (0.9)	4.3 (0.8)	3.7 (1.1)	4.5 (0.9)
Gain	1.5 (1.2)	2.0 (1.9)	1.2 (1.4)	2.8 (1.1)
Pen held	13.4 (7.9)	11.0 (7.7)	13.7 (7.5)	15.6 (7.6)
Chat size	2604.7 (2359.0)	2726.4 (2470.9)	2818.3 (1473.6)	2494.8 (1935.2)
Changes	148.8 (69.2)	197.7 (105.2)	179.4 (110.9)	193.5 (75.5)

3 Conclusion

We presented a study comparing the effects of domain and collaboration feedback on learning within COLLECT-*UML*. All four treatment conditions improved significantly between pre- and post-test, showing that practicing within COLLECT-*UML* helps learning. At a surface level there was no difference in collaboration between modes. The DCF mode learnt significantly more than other modes, indicating the value of receiving both domain and collaborative feedback. Surprisingly, the CF mode had the lowest learning gains (lower than NF). One possible interpretation of this could be without domain advice students simply shared and possibly even promoted their misconceptions. We plan to perform deeper analyses of collaboration quality and problem-solving progress.

References

- [1] Tchounikine, P., Rummel, N., McLaren, B., "Computer Supported Collaborative Learning and Intelligent Tutoring Systems," in R. Nkambou et al. (Eds.) *Advances in Intelligent Tutoring Systems*. vol. 308, Berlin Heidelberg: Springer-Verlag, 2010, pp. 447–463.
- [2] Hausmann, R., Sande, B. v. d., VanLehn, K., "Shall We Explain? Augmenting Learning from Intelligent Tutoring Systems and Peer Collaboration," in B.P.Woolf et al. (Eds.) *Proc. Intelligent Tutoring Systems*. vol. 5091, Springer-Verlag, 2008, pp. 636–645.
- [3] Baghaei, N. and Mitrovic, A., "From Modelling Domain Knowledge to Metacognitive Skills: Extending a Constraint-Based Tutoring System to Support Collaboration," in C. Conati et al. (Eds.) *Proc. User Modeling*. vol. 4511, Springer-Verlag, 2007, pp. 217–227.
- [4] Baghaei, N., Mitrovic, A., and Irwin, W., "Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams," *Computer-Supported Collaborative Learning*, 2, 159–190, 2007.